Summer 2022 Data Science Intern Challenge

Question 1: Given some sample data, write a program to answer the following: click here (https://docs.google.com/document/d/13VCtoyto9X1PZ74nPI4ZEDdb8hF8LAlcmLH1ZTHxKxE/edit#) to access the required data set

On Shopify, we have exactly 100 sneaker shops, and each of these shops sells only one model of shoe. We want to do some analysis of the average order value (AOV). When we look at orders data over a 30 day window, we naively calculate an AOV of $3145.13. Given that we know these shops are selling sneakers, a relatively affordable item, something seems wrong with our analysis.

1. Think about what could be going wrong with our calculation. Think about a better way to evaluate this data.
2. What metric would you report for this dataset?
3. What is its value?

1. Think about what could be going wrong with our calculation. Think about a better way to evaluate this data. Answer:

Let's import pandas library, read data file and explore the data.

```
In [2]: import pandas as pd
```

```
In [3]: Data = pd.read_excel('2019 Winter Data Science Intern Challenge Data Set.xlsx')
```

Looking at a first few rows of the data:

In [4]:
```
Data.head()
```

Out[4]:

	order_id	shop_id	user_id	order_amount	total_items	payment_method	created_at
0	1	53	746	224	2	cash	2017-03-13 12:36:56.190
1	2	92	925	90	1	cash	2017-03-03 17:38:51.999
2	3	44	861	144	1	cash	2017-03-14 04:23:55.595
3	4	18	935	156	1	credit_card	2017-03-26 12:43:36.649
4	5	18	883	156	1	credit_card	2017-03-01 04:35:10.773

Understanding the data better by looking at data descriptive statistics such as mean, standard deviation min and max of each column.

In [5]:
```
Data.describe()
```

Out[5]:

	order_id	shop_id	user_id	order_amount	total_items
count	5000.000000	5000.000000	5000.000000	5000.000000	5000.00000
mean	2500.500000	50.078800	849.092400	3145.128000	8.78720
std	1443.520003	29.006118	87.798982	41282.539349	116.32032
min	1.000000	1.000000	607.000000	90.000000	1.00000
25%	1250.750000	24.000000	775.000000	163.000000	1.00000
50%	2500.500000	50.000000	849.000000	284.000000	2.00000
75%	3750.250000	75.000000	925.000000	390.000000	3.00000
max	5000.000000	100.000000	999.000000	704000.000000	2000.00000

Let's check how many uniques shops are available in the dataset using pandas *nunique()* function. There are 100 unique shops in the data.

In [6]:
```
Data["shop_id"].nunique()
```

Out[6]: 100

It seems that the naive AOV calculation is obtained by taking the average over all **order_amount** values. The mean across the _order*amount* column is calculated in below using pandas *mean()* function and you can see that this value is equal to the naive AOV calculation mentioned in the question.

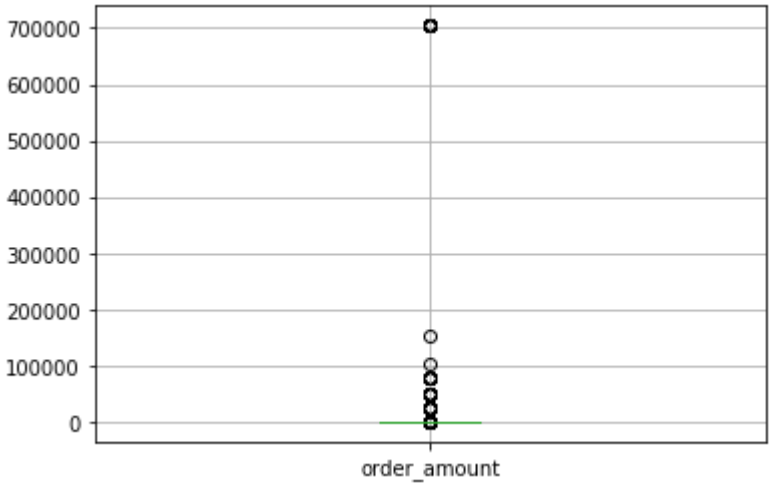
In [7]:
```
Naive_AOV = Data["order_amount"].mean()
Naive_AOV
```

Out[7]: 3145.128

As the mean taken over the order_amount column is pretty high for a pair of sneakers, there may be outliers in this column since mean is sensitive to outliers.

In [8]:
```
Data.boxplot(column=['order_amount'])
```

Out[8]: <matplotlib.axes._subplots.AxesSubplot at 0x2bb9f0ff948>



Therefore, we took a look at this column distribution by calling the boxplot function that makes a box plot from DataFrame columns. By looking at the boxplot of the **order_amount** column, we can see that there are outliers in this column (for example there is an order_amount of about 700000!!). Let's see how many rows of these outliers exist in the dataset.

In [9]:
```
Data.groupby(['order_amount']).count().sort_values(by='order_amount', ascending=False)
```

Out[9]:

order_amount	order_id	shop_id	user_id	total_items	payment_method	created_at
704000	17	17	17	17	17	17
154350	1	1	1	1	1	1
102900	1	1	1	1	1	1
77175	9	9	9	9	9	9
51450	16	16	16	16	16	16
...	...	...	...	...	...	...
112	48	48	48	48	48	48
111	16	16	16	16	16	16
101	15	15	15	15	15	15
94	25	25	25	25	25	25
90	18	18	18	18	18	18

258 rows × 6 columns

There are high order_amounts that accoured multiple times (order amount 704000 occured 17 times, order amount 77175 occured 9 times, ...). This is the reason that the mean is skewed.

2. What metric would you report for this dataset? Answer:

One option can be calculating the median (which is not sensitive to outliers) of the order_amount column which is 284.

In [36]:
```
Median = Data['order_amount'].median()
Median
```

Out[36]: 284.0

Another approach is removing outliers and then calculating the mean of order amounts. A data point in a distribution is considered as an outlier if it is more than 1.5 times the length of the box away from either the lower or upper quartiles. Specifically, if a number is less than **Q1 – 1.5 × IQR** or greater than **Q3 + 1.5 × IQR**, then it is called an outlier. In function _Remove_Outlier*Indices* in below, a dataframe is given to the function and the functions returns the indices of data points after outlier removal.

In [15]:
```
# Removing Outliers calculation

def Remove_Outlier_Indices(df):
    Q1 = df.quantile(0.25)
    Q3 = df.quantile(0.75)
    IQR = Q3 - Q1
    trueList = ~((df< (Q1 - 1.5 * IQR)) |(df > (Q3 + 1.5 * IQR)))
    return trueList
```

In [37]:
```
nonOutlierList = Remove_Outlier_Indices(Data['order_amount'])
# nonOutlierData stores the data obtained after removing outliers
nonOutlierData = Data[nonOutlierList]
nonOutlierData.describe()
```

Out[37]:

	order_id	shop_id	user_id	order_amount	total_items
count	4859.000000	4859.000000	4859.000000	4859.000000	4859.000000
mean	2497.395966	49.852645	849.905742	293.715374	1.950196
std	1443.356555	29.049171	86.887496	144.453395	0.919791
min	1.000000	1.000000	700.000000	90.000000	1.000000
25%	1244.500000	24.000000	776.000000	162.000000	1.000000
50%	2498.000000	50.000000	850.000000	280.000000	2.000000
75%	3749.500000	74.000000	925.000000	380.000000	3.000000
max	5000.000000	100.000000	999.000000	730.000000	5.000000

The mean of the order_amount after removing outlirs is about $294.

Question 2: For this question you'll need to use SQL. Follow this link (https://www.w3schools.com/SQL/TRYSQL.ASP?FILENAME=TRYSQL_SELECT_ALL) to access the data set required for the challenge. Please use queries to answer the following questions. Paste your queries along with your final numerical answers below.

How many orders were shipped by Speedy Express in total?

```
SELECT COUNT(*) as NumberOfOrders

FROM Orders as O

JOIN Shippers as S

ON O.ShipperID = S.ShipperID

WHERE ShipperName = "Speedy Express"
```

Output: 54

What is the last name of the employee with the most orders?

```
SELECT E.LastName, COUNT(OrderID) as NumberOfOrders
FROM Employees as E
JOIN Orders as O
on E.EmployeeID = O.EmployeeID
GROUP BY E.EmployeeID
ORDER BY NumberOfOrders DESC
LIMIT 1;
```

Output: Peacock with 40 orders

What product was ordered the most by customers in Germany?

```
SELECT ProductName, SUM(OD.Quantity) as NumberOfOrders
FROM Orders as O
JOIN OrderDetails as OD ON O.OrderID = OD.OrderID
JOIN Customers as C ON C.CustomerID = O.CustomerID
JOIN Products as P ON P.ProductID = OD.ProductID
WHERE Country = "Germany"
GROUP BY Country, ProductName
ORDER BY NumberOfOrders DESC
LIMIT 1;
```

Output: 160 orders of Boston Crab Meat